Autenticação e Debug PPP

CISCO ACADEMY - NCE/UFRJ

ppp authentication [pap | pap chap | chap pap | chap] callin

1. pap

A autenticação pap não precisaria de autenticação bi-direcional como está no exemplo abaixo.

```
Lab_A#config terminal
Lab_A(config)#username Lab_B password cisco

Lab_A(config)#interface serial 0
Lab_A(config-if)#encapsulation ppp
Lab_A(config-if)#ppp authentication pap
Lab_A(config-if)#ppp pap sent-username Lab_A password cisco

Lab_A#debug ppp authentication
```

```
Lab_B#config terminal
Lab_B(config)#username Lab_A password cisco

Lab_B(config)#interface serial 0
Lab_B(config-if)#encapsulation ppp
Lab_B(config-if)#ppp authentication pap
Lab_B(config-if)#ppp pap sent-username Lab_B password cisco

Lab_B#debug ppp authentication
```

a. Autenticação com Sucesso

```
Se0 PPP: Treating connection as a dedicated line
Se0 PAP: O AUTH-REQ id 197 len 16 from "Lab_A"
Se0 PAP: I AUTH-REQ id 197 len 16 from "Lab_B"
Se0 PAP: Authenticating peer Lab_B
Se0 PAP: O AUTH-ACK id 197 len 5
Se0 PAP: I AUTH-ACK id 197 len 5
```

b. Autenticação sem Sucesso (usuário não existente)

```
Se0 PPP: Treating connection as a dedicated line
Se0 PAP: O AUTH-REQ id 1 len 17 from "Lab_A"
Se0 PAP: I AUTH-REQ id 1 len 16 from "Lab_X"
Se0 PAP: Authenticating peer Lab_X
Se0 PAP: O AUTH-NAK id 1 len 27 msg is "Authentication failure" Username Lab_X not found
```


c. Autenticação sem Sucesso (senha diferente)

```
Se0 PPP: Treating connection as a dedicated line
Se0 PAP: O AUTH-REQ id 19 len 17 from "Lab_A"
Se0 PAP: I AUTH-REQ id 19 len 16 from "Lab_B"
Se0 PAP: Authenticating peer Lab_B
Se0 PAP: O AUTH-ACK id 19 len 5
Se0 PAP: I AUTH-NAK id 19 len 27 msg is "Authentication failure"
```

2. chap

Se opção "callin" não for usada a autenticação chap precisa de autenticação bi-direcional. Se "callin" for usada (no roteador que faz a chamada) apenas o roteador que recebe a chamada faz a autenticação.

```
Lab_A#config terminal
Lab_A(config)#username Lab_B password cisco

Lab_A(config)#interface serial 0
Lab_A(config-if)#encapsulation ppp
Lab_A(config-if)#ppp authentication chap

Lab_A#debug ppp authentication
```

```
Lab_B#config terminal
Lab_B(config)#username Lab_A password cisco

Lab_B(config)#interface serial 0
Lab_B(config-if)#encapsulation ppp
Lab_B(config-if)#ppp authentication chap

Lab_B#debug ppp authentication
```

a. Autenticação com Sucesso

```
Se0 PPP: Treating connection as a dedicated line
Se0 CHAP: O CHALLENGE id 3 len 26 from "Lab_A"
Se0 CHAP: I CHALLENGE id 83 len 26 from "Lab_B"
Se0 CHAP: O RESPONSE id 83 len 26 from "Lab_A"
Se0 CHAP: I RESPONSE id 3 len 26 from "Lab_B"
Se0 CHAP: O SUCCESS id 3 len 4
Se0 CHAP: I SUCCESS id 83 len 4
```

b. Autenticação sem Sucesso (senha diferente)

```
Se0 PPP: Treating connection as a dedicated line
Se0 CHAP: O CHALLENGE id 24 len 26 from "Lab_A"
Se0 CHAP: I CHALLENGE id 104 len 26 from "Lab_B"
Se0 CHAP: O RESPONSE id 104 len 26 from "Lab_A"
Se0 CHAP: I RESPONSE id 24 len 26 from "Lab_B"
Se0 CHAP: O FAILURE id 24 len 25 msg is "MD/DES compare failed"
```

Configuring PPP for Wide-Area Networking

The Point-to-Point Protocol (PPP), described in RFCs 1661 and 1332, encapsulates network layer protocol information over point-to-point links. You can configure PPP on the following types of physical interfaces:

- Asynchronous serial
- HSSI
- ISDN
- Synchronous serial

By enabling PPP encapsulation on physical interfaces, PPP can also be in effect on calls placed by the dialer interfaces that use the physical interfaces.

The current implementation of PPP supports option 3, authentication using CHAP or PAP, option 4, Link Quality Monitoring, and option 5, Magic Number configuration options. The software always sends option 5 and negotiates for options 3 and 4 if so configured. All other options are rejected.

Cisco supports the following upper-layer protocols: AppleTalk, Bridging, CLNS, DECnet, IP, IPX, VINES, and XNS.

The software provides PPP as an encapsulation method. It also provides the Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) on serial interfaces running PPP encapsulation. The following sections describe the tasks to configure PPP routing features.

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the **down-when-looped** command is configured, the router might shut down a link if it detects a loop.

PPP Configuration Task List

To configure PPP on a serial interface, perform the following task in interface configuration mode:

- Enable PPP Encapsulation

You can also complete the tasks in the following sections; these tasks are optional but offer a variety of uses and enhancements for PPP on your systems and networks:

- Enable CHAP or PAP Authentication
- Enable Link Quality Monitoring (LQM)
- Configure Automatic Detection of Encapsulation Type
- Configure Compression of PPP Data

- Configure IP Address Pooling
- Configure PPP Callback
- Disable or Reenable Peer Neighbor Routes
- Configure PPP Half-Bridging
- Configure Multilink PPP
- Configure Multichassis Multilink PPP
- Configure Virtual Private Dial-up Networks
- Enable PPP on VTY Lines for Asynchronous Access over ISDN

See the “PPP Examples” section at the end of this chapter.

Enable PPP Encapsulation

You can enable PPP on serial lines to encapsulate IP and other network protocol datagrams. To do so, perform the following task in interface configuration mode:

Task	Command
Enable PPP encapsulation.	**encapsulation ppp**

PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. The **no keepalive command** can be used to disable echo requests.

Enable CHAP or PAP Authentication

The Point-to-Point Protocol (PPP) with Challenge Handshake Authentication Protocol (CHAP) authentication or Password Authentication Protocol (PAP) is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP are specified in RFC 1334. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a *name*. This identification process prevents a router from placing another call to a router to which it is already connected, and also prevents unauthorized access. See the “Configuring Interfaces” chapter in the *Configuration Fundamentals Configuration Guide* for more information about CHAP and PAP.

Access control using Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

The required response consists of two parts:

- An encrypted version of the ID, a secret password (or *secret*), and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret by performing the same encryption operation as indicated in the response and looking up the required host name or username. The secret passwords must be identical on the remote device and the local router.

By transmitting this response, the secret is never transmitted in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only at the time a link is established. The local router or access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

To use CHAP or PAP, you must perform the following tasks:

Step 1 Enable PPP encapsulation.

Step 2 Enable CHAP or PAP on the interface.

Step 3 For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

To enable PPP encapsulation, perform the following task in interface configuration mode:

Task	Command
Enable PPP on an interface.	**encapsulation ppp**

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, perform the following task in interface configuration mode:

Task	Command
Define the authentication methods supported and the order in which they are used.	**ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**]

The **ppp authentication chap** optional keyword **if-needed** can be used only with TACACS or extended TACACS. The optional keyword *list-name* can only be used with AAA/TACACS+.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

Add a **username** entry for each remote system from which the local router or access server requires authentication.

To specify the password to be used in CHAP or PAP caller identification, perform the following task in global configuration mode:

Task	Command
Configure identification.	**username** *name* **password** *secret*

To configure Terminal Access Controller Access Control System (TACACS) on a specific interface as an alternative to global host authentication, perform the following task in interface configuration mode:

Task	Command
Configure TACACS.	**ppp use-tacacs** [**single-line**][1] or **aaa authentication ppp**[2]

1. This command is documented in the "System Management Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "Network Access Security Commands" chapter in the *Security Command Reference*.

Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with Authentication, Authorization, and Accounting (AAA)/TACACS+.

For an example of CHAP, see the section "CHAP with an Encrypted Password Examples" at the end of this chapter. CHAP and PAP are specified in RFC 1334, "The PPP Authentication Protocols," by Brian Lloyd of Lloyd and Associates and William A. Simpson of Computer Systems Consulting Services.

Enable Link Quality Monitoring (LQM)

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the router shuts down the link. The percentages are calculated for both the incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent with the total number of packets and bytes received by the destination node. The incoming quality is calculated by comparing the total number of packets and bytes received with the total number of packets and bytes sent by the destination peer.

When LQM is enabled, Link Quality Reports (LQRs) are sent every keepalive period. LQRs are sent in place of keepalives. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

LQR is specified in RFC 1333, "PPP Link Quality Monitoring," by William A. Simpson of Computer Systems Consulting Services.

To enable LQM on the interface, perform the following task in interface configuration mode:

Task	Command
Enable LQM on the interface.	**ppp quality** *percentage*

The *percentage* argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and taken down.

Configure Automatic Detection of Encapsulation Type

You can enable a serial or ISDN interface to accept calls and dynamically change the encapsulation in effect on the interface when the remote device does not signal the call type. For example, if an ISDN call does not identify the call type in the Lower Layer Compatibility fields and is using an encapsulation that is different from the one configured on the interface, the interface can change its encapsulation type on the fly.

This feature enables interoperation with ISDN terminal adapters that use V.120 encapsulation but do not signal V.120 in the call setup message. An ISDN interface that by default answers a call as synchronous serial with PPP encapsulation can change its encapsulation and answer such calls.

Automatic detection is attempted for the first 10 seconds after the link is established or the first five packets exchanged over the link, whichever is first.

To enable automatic detection of encapsulation type, perform the following task in interface configuration mode:

Task	Command
Enable automatic detection of encapsulation type on the specified interface.	**autodetect encapsulation** *encapsulation-type*

You can specify one or more encapsulations to detect. Cisco IOS software currently supports automatic detection of PPP and V.120 encapsulations.

Configure Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. The compression algorithm used is a predictor algorithm (the RAND algorithm), which uses a compression dictionary to predict the next character in the frame.

PPP encapsulations support both predictor and Stacker compression algorithms.

Compression is performed in software and might significantly affect system performance. Cisco recommends that you disable compression if the router CPU load exceeds 65 percent. To display the CPU load, use the **show process cpu** EXEC command.

If the majority of your traffic is already compressed files, do not use compression.

To configure compression over PPP, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable encapsulation of a single protocol on the serial line.	**encapsulation ppp**
Step 2	Enable compression.	**ppp compress** [**predictor** | **stac**]

Configure IP Address Pooling

Point-to-point interfaces must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be obtained from a variety of sources. The address can be configured through the command line, entered with an EXEC-level command or provided by TACACS+, DHCP, or from a locally administered pool.

IP address pooling consists of a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through the IP Control Protocol (IPCP) address negotiation process. It also enhances the flexibility of configuration by allowing multiple types of pooling to be active simultaneously.

The IP address pooling feature now allows the configuration of a global default address pooling mechanism, per-interface configuration of the mechanism, and per-interface configuration of a specific address or pool name.

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup—This method is used only if the peer requests an IP address, no other peer IP address has been assigned, and the interface is a member of a dialer group.
- PPP or SLIP EXEC command—An asynchronous dial-up user can enter a peer IP address or host name when PPP or SLIP is invoked from the command line. The address is used for the current session and then discarded.
- IPCP negotiation—If the peer presents a peer IP address during IPCP address negotiation and no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Chat script—The IP address in the dialer map command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- VTY/Protocol translation—The translate command can define the peer IP address for a VTY (pseudo async interface).
- Default IP address—The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.
- TACACS+ assigned IP address—During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that the user being authenticated on a dial-up interface can use. This address overrides any default IP address and prevents pooling from taking place.
- DHCP retrieved IP address—If configured, the routers acts as a proxy client for the dial-up user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.
- Local address pool—The local address pool contains a set of contiguous IP addresses (a maximum of 256 addresses) stored in two queues. The *free* queue contains addresses available to be assigned and the *used* queue contains addresses that are in use. Addresses are stored to the free queue in first-in first-out (FIFO) order to minimize the chance the address will be reused and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.

  The pool configured for the interface is used, unless TACACS+ returns a pool name as part of authentication, authorization, and accounting (AAA). If no pool is associated with a given interface, the global pool named *default* is used.

Precedence Rules

The following precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address exists)
4. Address from an EXEC-level PPP or SLIP command or from a chat script
5. Configured address from the **peer default ip address** command or address from the protocol **translate** command
6. Peer provided address from IPCP negotiation (not accepted unless no other address exists)

Interfaces Affected

This feature is available on all asynchronous serial, synchronous serial, ISDN BRI, and ISDN PRI interfaces running the Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP).

Choose the IP Address Assignment Method

The IP address pooling feature now allows configuration of a global default address pooling mechanism, per-interface configuration of the mechanism, and per-interface configuration of a specific address or pool name.

You can define the type of IP address pooling mechanism used on router interfaces in the following ways:

- Define the Global Default Mechanism
- Configure Per-Interface IP Address Assignment

Define the Global Default Mechanism

The global default mechanism applies to all point-to-point interfaces (asynchronous, synchronous, ISDN BRI, ISDN PRI, and dialer interfaces) that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the global default mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform the tasks in one of following sections:

- Define DHCP as the Global Default Mechanism
- Define Local Address Pooling as the Global Default Mechanism

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface or you can configure the interface with a specific IP address to be used for dial-in peers.

Define DHCP as the Global Default Mechanism

The Dynamic Host Configuration Protocol (DHCP) specifies the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy-client—A Cisco access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. The DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

To enable DHCP as the global default mechanism, complete the following tasks in global configuration mode:

Task		Command
Step 1	Specify DHCP client-proxy as the global default mechanism.	**ip address-pool dhcp-proxy-client**
Step 2	(Optional) Specify the IP address of a DHCP server for the proxy client to use.	**ip dhcp-server** [*ip-address* | *name*]

In Step 2, you can provide as few as one or as many as ten DHCP servers for the proxy-client (the Cisco router or access server) to use. DHCP servers provide temporary IP addresses.

Define Local Address Pooling as the Global Default Mechanism

To specify that the global default mechanism to use is local pooling, complete the following tasks in global configuration mode:

Task		Command
Step 1	Specify local pooling as the global default mechanism.	**ip address-pool local**
Step 2	Create one or more local IP address pools.	**ip local pool** {**default** | *poolname*} *low-ip-address* [*high-ip-address*]

If no other pool is defined, the local pool called *default* is used.

Configure Per-Interface IP Address Assignment

When you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can then configure the few interfaces for which it is important to have a nondefault configuration. You can do any of the following;

- Define a nondefault address pool for use by a specific interface.
- Define DHCP on an interface even if you have defined local pooling as the global default mechanism.
- Specify one IP address to be assigned to all dial-in peers on an interface.
- Make temporary IP addresses available on a per-interface basis to asynchronous clients using Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP).

To define a nondefault address pool for use on an interface, perform the following tasks beginning in global configuration mode:

Task	Command
Create one or more local IP address pools.	**ip local pool** *poolname* {*low-ip-address* [*high-ip-address*]}
Specify the interface and enter interface configuration mode.	**interface** *type number*
Specify the pool for the interface to use.	**peer default ip address pool** *poolname*

To define DHCP as the IP address mechanism for an interface, complete the following tasks beginning in global configuration mode:

Task	Command
Specify the interface and enter interface configuration mode.	**interface** *type number*
Specify DHCP as the IP address mechanism on this interface.	**peer default ip address pool dhcp**

To define a specific IP address to be assigned to all dial-in peers on an interface, complete the following tasks beginning in global configuration mode:

Task	Command
Specify the interface and enter interface configuration mode.	**interface** *type number*
Specify the IP address to assign.	**peer default ip address** *ip-address*

To make temporary IP addresses available on a per-interface basis for dial-in asynchronous clients using Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP), perform the following tasks, beginning in global configuration mode:

Task		Command
Step 1	Specify that the access server use a local IP address pool on all asynchronous interfaces.	**ip address-pool local**
Step 2	Create one or more local IP address pools.	**ip local pool** {**default** | *poolname* {*begin-ip-address-range* [*end-ip-address-range*]}}
Step 3	(Optional) Enter interface configuration mode.	**interface async** *number*
Step 4	(Optional) If you want an interface to use an address pool other than default, specify which pool each interface uses.	**peer default ip address pool** *poolname*

Configure PPP Callback

PPP callback provides a client-server relationship between the end points of a point-to-point connection. PPP callback allows a router to request that a dial-up peer router call back. The callback feature can be used to control access and toll costs between the routers.

When PPP callback is configured on the participating routers, the calling router (the callback client) passes authentication information to the remote router (the callback server), which uses the host name and dial string authentication information to determine whether to place a return call. If the authentication is successful, the callback server disconnects and then places a return call. The remote username of the return call is used to associate it with the initial call so that packets can be transmitted.

Both routers on a point-to-point link must be configured for PPP callback; one must function as a callback client and one must be configured as a callback server.. The callback client must be configured to initiate PPP callback, and the callback server must be configured to accept PPP callback.

This feature implements the following callback specifications of RFC 1570:

- For the client—Option 0, location is determined by user authentication
- For the server—Option 0, location is determined by user authentication; Option 1, dialing string; and Option 3, E.164 number.

Return calls are made through the same dialer rotary group but not necessarily the same line as the initial call.

Note If the return call fails (because the line is not answered or the line is busy), no retry occurs. If the callback server has no interface available when attempting the return call, it does not retry.

For an example of configuring PPP callback, see the "PPP Callback Example" section later in this chapter.

Configure a Router as a Callback Client

To configure a router interface as a callback client, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify the interface.	**interface serial** *number*[1]
Step 2	Enable DDR. Set parity on synchronous serial interfaces and asynchronous interfaces.	**dialer in-band** [**no-parity** | **odd-parity**] [2]
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	Enable CHAP or Password Authentication Protocol (PAP) authentication.	**ppp authentication chap** or **ppp authentication pap**
Step 5	Map the next hop address to the host name and phone number.	**dialer map** *protocol next-hop-address* **name** *hostname dial-string* [2]
Step 6	Enable the interface to request PPP callback for this callback map class.	**ppp callback request**
Step 7	Configure a dialer hold queue to store packets for this callback map class. (Optional)	**dialer hold-queue** *packets* **timeout** *seconds* [2]

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "DDR Commands" chapter of this manual.

Configure a Router as a Callback Server

To configure a router as a callback server, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify the interface and enter interface configuration mode.	**interface serial** *number*[1]
Step 2	Enable DDR. Set parity on synchronous serial interfaces and asynchronous interfaces.	**dialer in-band** [**no-parity** | **odd-parity**] [2]
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	Enable CHAP or PAP authentication.	**ppp authentication** {**chap** | **pap**}
Step 5	Map the next hop address to the host name and phone number, using the name of the map-class established for PPP callback on this interface.	**dialer map** *protocol address* **name** *hostname* **class** *classname dial-string* [2]
Step 6	Configure a dialer hold queue to store packets to be transferred when the callback connection is established. (Optional)	**dialer hold-queue** *number* **timeout** *seconds* [2]
Step 7	Configure a timeout period between calls (Optional).	**dialer enable-timeout** *seconds* [2, 3]
Step 8	Configure the interface to accept PPP callback.	**ppp callback accept**
Step 9	Enable callback security, if desired. (Optional)	**dialer callback-secure**
Step 10	Return to global configuration mode.	**exit**[4]
Step 11	Configure a dialer map class for PPP callback.	**map-class dialer** *classname*
Step 12	Configure a dialer map class as a callback server.	**dialer callback-server** [**username**]

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "DDR Commands" chapter of this manual.
3. Default is 15 seconds for enable timer. Time between the initial call and the return call can be improved by reducing this number, but care should be taken to ensure that the initial call is completely disconnected before the timer expires.
4. This command is documented in the "Image and Configuration File Load Commands" chapter of the *Configuration Fundamentals Command Reference*.

Note On the PPP callback server, the **dialer enable-timeout** functions as the timer for returning calls to the callback client.

Disable or Reenable Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.

To disable this default behavior or to reenable it once it has been disabled, complete the following tasks in interface configuration mode:

Task	Command
Disable creation of neighbor routes.	**no peer neighbor-route**
Reenable creation of neighbor routes.	**peer neighbor-route**

Note If entered on a dialer or async-group interface, this command affects all member interfaces.

Configure PPP Half-Bridging

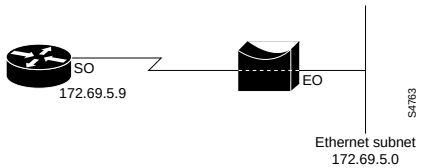
For situations in which a routed network needs connectivity to a remote bridged Ethernet network, a serialor ISDN interface can be configured to function as a PPP half-bridge. The line to the remote bridge functions as a virtual Ethernet interface, and the router's serial or ISDN interface functions as a node on the same Ethernet subnetwork as the remote network.

The bridge sends bridge packets to the PPP half-bridge, which converts them to routed packets and forwards them to other router processes. Likewise, the PPP half-bridge converts routed packets to Ethernet bridge packets and sends them to the bridge on the same Ethernet subnetwork.

Note An interface cannot function as both a half-bridge and a bridge.

Figure 37 shows a router with a serial interface configured as a PPP half-bridge. The interface functions as a node on the Ethernet subnetwork with the bridge. Note that the serial interface has an IP address on the same Ethernet subnetwork as the bridge.

Figure 37 Router Serial Interface Configured as a Half-Bridge



Note The Cisco IOS software supports no more than one PPP half-bridge per Ethernet subnetwork.

To configure a serial interface to function as a half-bridge, compete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify the interface (and enter interface configuration mode).	**interface serial** *number* [1]
Step 2	Enable PPP half-bridging for one or more routed protocols: AppleTalk, IP, or IPX.	**ppp bridge appletalk** **ppp bridge ip** **ppp bridge ipx** [**novell-ether** | **arpa** | **sap** | **snap**]
Step 3	Provide a protocol address on the same subnetwork as the remote network.	**ip address** *n.n.n.n* [2] **appletalk address** *network.node* [3] **appletalk cable-range** *cable-range network.node* [3] **ipx network** *network*

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "IP Commands" chapter in the *Network Protocols Command Reference, Part 1*.
3. This command is documented in the "AppleTalk Commands" chapter in the *Network Protocols Command Reference, Part 2*.

Note You must enter the **ppp bridge** command either when the interface is shut down or before you provide a protocol address for the interface.

For more information about AppleTalk addressing see the "Configuring AppleTalk" chapter; for more information about IPX addresses and encapsulations, see the "Configuring Novell IPX" chapter. Both chapters are in the *Network Protocols Configuration Guide, Part 2*.

Configure Multilink PPP

The Multilink Point-to-Point Protocol (PPP) feature provides load balancing functionality over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic. Cisco's implementation of Multilink PPP supports the fragmentation and packet sequencing specifications in RFC 1717.

Multilink PPP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a dialer load threshold that you define. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

Multilink PPP is designed to work over single or multiple interfaces of the following types that are configured to support both dial-on-demand rotary groups and PPP encapsulation:

- Asynchronous serial interfaces
- Basic Rate Interfaces (BRIs)
- Primary Rate Interfaces (PRIs)

Configure Multilink PPP on Asynchronous Interfaces

To configure Multilink PPP on asynchronous interfaces, you configure the asynchronous interfaces to support DDR and PPP encapuslation, then you configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

To configure an asynchronous interface to support DDR and PPP encapsulation, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify an asynchronous interface.	**interface async** *number* [1]
Step 2	Specify no IP address for the interface.	**no ip address**
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	Enable DDR on the interface.	**dialer in-band** [2]
Step 5	Include the interface in a specific dialer rotary group.	**dialer rotary-group** *number* [2]

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "DDR Commands" chapter of this manual.

Repeat this step for additional asynchronous interfaces, as needed.

At some point, adding more asynchronous interfaces does not improve performance, With the default MTU size, Multilink PPP should support three asynchronous interfaces using V.34 modems. However, packets might be dropped occasionally if the MTU is small or large bursts of short frames occur.

To configre a dialer interface to support PPP encapsulation and Multilink PPP, complete the following tasks beginning in global configuration mode:

Task		Command
Step 1	Define a dialer rotary group.	**interface dialer** *number* [1]
Step 2	Specify no IP address for the interface.	**no ip address**
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	Enable DDR on the interface.	**dialer in-band** [1]
Step 5	Configure bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.	**dialer load-threshold** *load* [**inbound** | **outbound** | **either**] [1]
Step 6	Enable Multilink PPP.	**ppp multilink**

1. This command is documented in the "DDR Commands" chapter of this manual.

Configure Multilink PPP on a Single ISDN BRI Interface

To enable Multilink PPP on a single Integrated Services Digital Network (ISDN) BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

To enable PPP on an ISDN BRI interface, perform the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify an interface.	**interface bri** *number*[1]
Step 2	Provide an appropriate protocol address for the interface.	**ip address** *ip-address mask*[2]
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	(Optional) Specify a dialer idle timeout.	**dialer idle-timeout** *seconds*
Step 5	Specify the dialer load threshold for bringing up additional WAN links.	**dialer load-threshold** *load*
Step 6	Configure the ISDN interface to call the remote site.	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
Step 7	Add the interface to a dialer rotary group.	**dialer-group** *group-number*
Step 8	(Optional) Enable PPP authentication.	**ppp authentication pap**
Step 9	Enable Multilink PPP on the dialer rotary group	**ppp multilink**

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "IP Commands" chapter in the *Network Protocols Command Reference, Part 1*.

If you do not use PPP authentication procedures (Step 8), your telephone service must pass caller ID information.

The load threshold number is required. For an example of configuring Multilink PPP on a single ISDN BRI interface, see the "One ISDN Interface Configured for Multilink PPP Example" section later in this chapter.

When Multilink PPP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer-load threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely and the **dialer-load threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Configure Multilink PPP on Multiple ISDN BRI Interfaces

To enable Multilink PPP on multiple ISDN BRI interfaces, you set up a dialer rotary interface and configure it for Multilink PPP and then you configure the BRIs separately and add them each to the same rotary group.

To set up the dialer rotary interface for the BRI interfaces, perform the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify the dialer rotary interface.	**interface dialer** *number* [1]
Step 2	Specify the protocol address for the dialer rotary interface.	**ip address** *address mask*[2]
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	Specify in-band dialing.	**dialer in-band** [3]

Task		Command
Step 5	(Optional) Specify the dialer idle timeout period, using the same timeout period as the individual BRI interfaces.	**dialer idle-timeout** *seconds* [3]
Step 6	Map the next-hop protocol address and name to the dial string needed to reach it.	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]] [3]
Step 7	Specify the dialer load threshold, using the same threshold as the individual BRI interfaces.	**dialer load-threshold** *load* [3]
Step 8	Control access to this interface by adding it to a dialer access group.	**dialer-group** *group-number* [3]
Step 9	(Optional) Enable PPP Challenge Handshake Authentication Protocol (CHAP) authentication.	**ppp authentication chap**
Step 10	Enable Multilink PPP.	**ppp multilink**

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "IP Commands" chapter in the *Network Protocols Command Reference, Part 1*.
3. This command is documented in the "DDR Commands" chapter in the *Wide-Area Networking Command Reference*

If you do not use PPP authentication procedures (Step 10), your telephone service must pass caller ID information.

To configure each of the BRIs to belong to the same rotary group, perform the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify one of the BRI interfaces.	**interface bri** *number*[1]
Step 2	Specify that it does not have an individual protocol address.	**no ip address**[2]
Step 3	Enable PPP encapsulation.	**encapsulation ppp**
Step 4	Set the dialer idle timeout period, using the same timeout for each of the BRI interfaces you configure.	**dialer idle-timeout** *seconds* [3]
Step 5	Add the interface to the rotary group.	**dialer rotary-group** *group-number* [3]
Step 6	Specify the dialer load threshold for bringing up additional WAN links.	**dialer load-threshold** *load* [3]

1. This command is documented in the "Interface Commands" chapter in the *Configuration Fundamentals Command Reference*.
2. This command is documented in the "IP Commands" chapter in the *Network Protocols Command Reference, Part 1*.
3. This command is documented in the "DDR Commands" chapter in the *Wide-Area Networking Command Reference.*

Repeat Steps 1 through 6 for each BRI you want to belong to the same dialer rotary group.

For an example of configuring Multilink PPP on multiple ISDN BRI interfaces, see the "Multiple ISDN Interfaces Configured for Multilink PPP Example" section later in this chapter.

When Multilink PPP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer load-threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely and the **dialer load-threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Configure Multichassis Multilink PPP

Prior to Release 11.2, Cisco IOS supported Multilink PPP. Beginning with Release 11.2, Cisco IOS software also supports MMP (MMP).

Multilink PPP provides the capability of splitting and recombining packets to a single end-system across a logical pipe (also called a *bundle*) formed by multiple links. Multilink PPP provides bandwidth on demand and reduces transmission latency across WAN links.

MMP, on the other hand, provides the additional capability for links to terminate at multiple routers with different remote addresses. MMP can also handle both analog and digital traffic.

The MMP feature is intended for situations with large pools of dialup users, for which the number of ports on a chassis cannot be allowed to be a limit. This feature allows companies to provide a single dialup number to its users and to apply the same solution to analog and digital calls. This feature allows internet service providers, for example, to allocate a single ISDN rotary number to several PRIs across several routers.

Multichassis Multilink PPP (MMP) does not require reconfiguration of telephone company switches.

Multichassis Multilink PPP is supported on the Cisco 7500, 4500, and 2500 series platforms and on synchronous serial, asynchronous serial, ISDN BRI, ISDN PRI, and dialer interfaces.

Understand Multichassis Multilink PPP

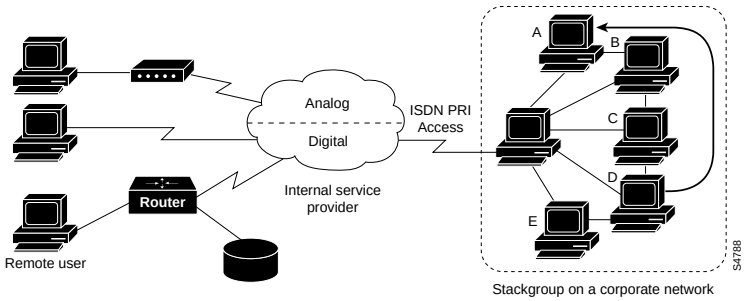
Routers or access servers are configured to belong to groups of peer routers, called *stack groups*. All members of the stack group are peers. Any stack group member can answer calls coming from a single access number, which is usually an ISDN PRI hunt group. Calls can come in from remote user devices, which can be routers, modems, ISDN terminal adapters, or PC cards.

Once a connection is established with one of the member of a stack group, that member owns the call. If a second call comes in from the same client and a different router answers the call, the router establishes a tunnel and forwards all packets belonging to the call to the router that owns the call.

With the availability of a more powerful router, it can be configured as a member of the stack group and the other stack group members can all establish tunnels and forward calls to it. In such a case, the other stack group members are just answering calls and forwarding traffic.

Note High-latency WAN lines between stack group members can make stack group operation inefficient.

Figure 38 Typical Multichassis Multilink PPP Scenario



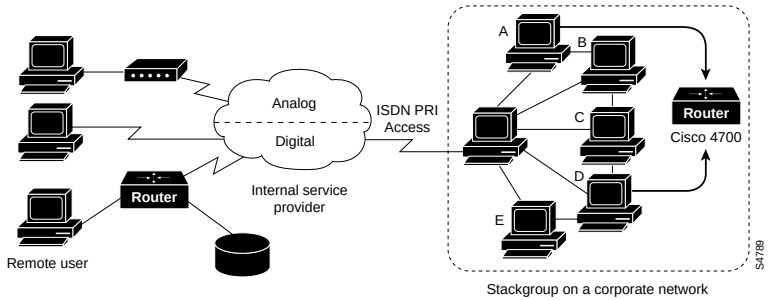
MMP call handling, bidding, and level-2 forwarding operation in the stack group proceeds as follows, as shown in Figure 38:

- When the first call coming in to the stack group, Router A answers.
- In the bidding, Router A wins because it already has the call. Router A becomes the *call-master* for that session with the remote device. (Router A might also be called the *host to the master bundle interface*.)
- When the remote device that initiated the call needs more bandwidth, it makes a second Multilink PPP call to the group.
- When the second call comes in Router D answers it and informs the stack group. Router A wins the bidding because it already is handling the session with that remote device.
- Router D establishes a tunnel to Router A, and forwards the raw PPP data to Router A.
- Router A reassembles and resequences the packets.
- If more calls come in to Router D and they too belong to Router A, the tunnel between A and D enlarges to handle the added traffic. Router D does not establish an additional tunnel to A.
- If more calls come in and are answered by any other router, that router also establishes a tunnel to A and forwards the raw PPP data.

The reassembled data is passed on the corporate network as if it had all come through one physical link.

In Figure 39, access servers that belong to a stack group answer calls, establish tunnels, and forward calls to a Cisco 4700 router that wins the bidding and is the call-master for all the calls. The Cisco 4700 reassembles and resequences all the packets coming in through the stack group.

Figure 39 Multichassis Multilink PPP with a Powerful Router as a Stack Group Member



Note You can build stack groups using different access server, switching, and router platforms. However, universal access servers such as the Cisco AS5200 should not be combined with ISDN-only access servers such as the 4*x*00 platform. Because calls from the central office are allocated in an arbitrary way, this combination could result in an analog call being delivered to a digital-only access server, which would not be able to handle the call.

Multichassis Multilink PPP (MMP) support on a group of routers requires that each router be configured to support the following:

- Multilink PPP
- Stack Group Bidding Protocol (SGBP)
- Virtual templates used for cloning interface configurations to support MMP

  A virtual template is a serial interface configuration with no hardware association.

Configure Multichassis Multilink PPP

To configure MMP, perform the steps in the following sections, in the order listed:

- Configure the Stack Group and Identify Members
- Configure a Virtual Template and Create a Virtual Template Interface

Configure the Stack Group and Identify Members

To configure the stack group on the router, complete the following steps beginning in global configuration mode:

Task		Command
Step 1	Create the stack group and assign this router to it.	**sgbp group** *group-name*
Step 2	Specify a peer member of the stack group. Repeat this step for each additional stack group peer.	**sgbp member** *peer-name* [*peer-ip-address*]

Note Only one stack group can be configured per access server or router.

Configure a Virtual Template and Create a Virtual Template Interface

To configure a virtual template for interfaces, perform the following tasks beginning in global configuration mode:

Task		Command
Step 1	Define a virtual template for the stack group.	**multilink virtual-template** *number*
Step 2	Specify an IP address pool by using any pooling mechanism—for example, IP local pooling or DHCP pooling.	**ip local pool default** *ip-address*
Step 3	Create a virtual template interface, and enter interface configuration mode.	**interface virtual-template** *number*
Step 4	If dialers are *not* configured on the physical interfaces, identify the virtual template interface type and number on the LAN.	**ip unnumbered ethernet 0**
Step 5	Enable PPP encapsulation on the virtual template interface.	**encapsulation ppp**
Step 6	Enable Multilink PPP on the virtual template interface.	**ppp multilink**
Step 7	Enable PPP authentication on the virtual template interface.	**ppp authentication chap**

If dialers are or will be configured on the physical interfaces, the **ip unnumbered** command, mentioned in Step 4, will be used in configuring the dialer interface. For examples that show MMP configured with and without dialers, see the "MMP Examples" at the end of this chapter.

For more information about address pooling, see the "Configure IP Address Pooling" section earlier in this chapter.

Configure Virtual Private Dial-up Networks

Virtual private dial-up networks allow separate and autonomous protocol domains to share common access infrastructure including modems, access servers, and ISDN routers. VPDN uses the Level 2 Forwarding protocol (L2F) which permits the tunneling of link level frames.

Using L2F tunneling, an Internet Service Provider (ISP) or other access service can create a virtual tunnel to link a customer's remote sites or remote users with corporate home networks. In particular, a network access server at the ISP's point of presence (POP) exchanges PPP messages with the remote users, and communicates by L2F requests and responses with the customer's home gateway to set up tunnels.

L2F passes protocol-level packets through the virtual tunnel between endpoints of a point-to-point connection.

Frames from the remote users are accepted by the ISP's POP, stripped of any linked framing or transparency bytes, encapsulated in L2F, and forwarded over the appropriate tunnel. The customer's home gateway accepts these L2F frames, strips the L2F encapsulation, and process the incoming frames for the appropriate interface.

Note This implementation of VPDN supports PPP dial-up only.

To configure virtual private dial-up networks, complete the tasks in the following sections:

- Understand Virtual Private Dial-up Networks
- Configure a Virtual Template and Create a Virtual Template Interface on the Home Gateway
- Configure Incoming VPDN Connections on the Home Gateway
- Configure Outgoing VPDN Connections on the Network Access Server

For more information, see the draft RFC *Level Two Forwarding (Protocol) "L2F"*, which describes the proposed implementation of L2F.

Understand Virtual Private Dial-up Networks

Virtual private dial-up networking enables users to configure secure networks that take advantage of Internet Service Providers that tunnel the company's remote access traffic through the ISP cloud.

Remote offices or mobile users can connect to their home network using local dial-up services of third parties. The dial-up service provider agrees to forward the company's traffic from the ISP POP to a company-run home gateway. Network configuration and security remains in the control of the client. The dial-up service provider provides a virtual pipe between the company's sites.

Note The MMP feature uses VPDN to connect multiple PPP sessions for which individual dial-in calls have arrived on different stack group members. VPDN provides speed and reliability for the setup and shutdown of Multilink PPP.

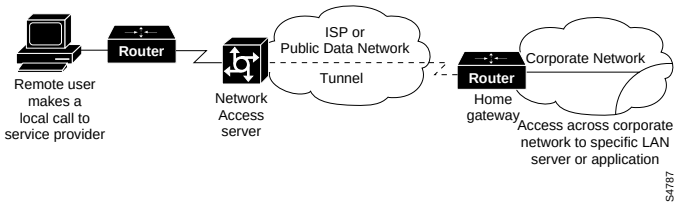
A VPDN connection between a remote user and the home LAN is done in the following steps:

1. The remote user initiates a PPP connection to the ISP using the analog telephone system or ISDN.
2. The ISP network access server accepts the connection.
3. The ISP network access server authenticates the end user with CHAP or PAP. The username is used to determine whether the user is an VPDN client. If the user is not a VPDN client, the client accesses the Internet or other contacted service.

4. The tunnel endpoints—the network access server and the home gateway—authenticate *each other* before any sessions are attempted within a tunnel.
5. If no L2F tunnel exists between the network access server and the remote users' home gateway, a tunnel is created. Once the tunnel exists, an unused slot within the tunnel is allocated.
6. The home gateway accepts or rejects the connection. Initial setup can include authentication information required to allow the home gateway to authenticate the user.
7. The home gateway sets up a virtual interface. Link-level frames can now pass through this virtual interface through the L2F tunnel.

Figure 40 illustrates a VPDN connection from a remote user, who makes a local call, to the corporate network, through an end-to-end L2F tunnel (shown by the dotted line).

Figure 40 Virtual Private Dialup Network



Configure a Virtual Template and Create a Virtual Template Interface on the Home Gateway

To configure a virtual template for interfaces on a home gateway access server, perform the following tasks beginning in global configuration mode:

Task		Command
Step 1	Specify a default local IP address pool.	**ip local pool default** *ip-address*
Step 2	Create a virtual template interface, and enter interface configuration mode.	**interface virtual-template** *number*
Step 3	Identify the virtual template interface type and number on the LAN.	**ip unnumbered ethernet 0**
Step 4	Enable PPP encapsulation on the virtual template interface.	**encapsulation ppp**
Step 5	Enable PPP authentication on the virtual template interface.	**ppp authentication chap**

Configure Incoming VPDN Connections on the Home Gateway

To configure virtual private dialup networking on a home gateway router or access server, complete the following tasks in global configuration mode:

Task	Command
Enable virtual private networking.	**vpdn enable**
Specify the remote host, the local name to use for authenticating, and the virtual template to use.	**vpdn incoming** *remote-name local-name* **virtual-template** *number*

Configure Outgoing VPDN Connections on the Network Access Server

To configure a network access server to make outgoing L2F connections to a home gateway for virtual private dialup networking, complete the following tasks in global configuration mode:

Task	Command
Enable virtual private networking.	**vpdn enable**
Specify the remote host that is to accept L2F connections.	**vpdn outgoing** *domain-name local-name* **ip** *ip-address*

Enable PPP on VTY Lines for Asynchronous Access over ISDN

You can configure a router to support asynchronous access over ISDN by globally enabling PPP on VTY lines. PPP is typically enabled on synchronous or asynchronous serial interfaces; however, the Cisco IOS software permits you to configure PPP on virtual terminal (VTY) lines. This configures the VTY line to support asynchronous access over ISDN from an ISDN terminal to a VTY session on the router.

To enable asynchronous protocol features on all the router's VTY lines, perform the following task in global configuration mode:

Task	Command
Configure all VTY lines to support asynchronous protocol features	**vty-async**[1]

1. This command is documented in the "Terminal Lines and Modem Support" chapter in the *Access Services Command Reference*.

This task enables PPP on VTY lines on a global basis on the router. To configure PPP on a per-VTY basis, use the **translate** command, which is documented in the "Protocol Translation Configuration Commands" chapter in the *Access Services Command Reference*.

Monitor and Maintain MLP, MMP, and VPDN Virtual Interfaces

To monitor and maintain virtual interfaces, you can perform any of the following tasks:

Task	Command
Display MLP and MMP bundle information.	**show ppp multilink**
Display information about the active L2F tunnels and the L2F message identifiers.	**show vpdn**
Display the status of the stack group members.	**show sgbp**
Display the current seed bid value.	**show sgbp queries**

PPP Examples

The examples provided in this section show various PPP configurations as follows:

- CHAP with an Encrypted Password Examples
- Multilink PPP Examples
- MMP Examples

- PPP Callback Example
- VPDN Examples

CHAP with an Encrypted Password Examples

The following configuration examples enable CHAP on interface serial 0 of three devices.

Configuration of Router yyy

```
hostname yyy
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

Configuration of Router xxx

```
hostname xxx
interface serial 0
encapsulation ppp
ppp authentication chap
username yyy password secretxy
username zzz password secretxz
```

Configuration of Router zzz

```
hostname zzz
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxz
username yyy password secretzy
```

When you look at the configuration file, the passwords will be encrypted and the display will look similar to the following:

```
hostname xxx
interface serial 0
encapsulation ppp
ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

Multilink PPP Examples

The following examples configure Multilink PPP. The first example configures it on one BRI interface, and the second configures multiple BRIs to belong to the same dialer rotary group, which is then configured for Multilink PPP.

Multilink PPP on One ISDN Interface Example

The following example enables Multilink PPP on the BRI 0 interface. Because an ISDN interface is a rotary group by default, when one BRI is configured, no dialer rotary group configuration is required.

```
interface bri 0
description connected to ntt 81012345678902
ip address 7.1.1.7 255.255.255.0
encapsulation ppp
dialer idle-timeout 30
dialer load-threshold 40 either
dialer map ip 7.1.1.8 name atlanta 81012345678901
dialer-group 1
ppp authentication pap
ppp multilink
```

Multilink PPP on Multiple ISDN Interfaces Example

The following example configures multiple ISDN BRIs to belong to the same dialer rotary group for Multilink PPP. The **dialer rotary-group** command is used to assign each of the ISDN BRIs to that dialer rotary group.

```
interface BRI0
no ip address
encapsulation ppp
dialer idle-timeout 500
dialer rotary-group 0
dialer load-threshold 255 balanced
!
interface BRI1
no ip address
encapsulation ppp
dialer idle-timeout 500
dialer rotary-group 0
dialer load-threshold 255 balanced
!
interface BRI2
no ip address
encapsulation ppp
dialer idle-timeout 500
dialer rotary-group 0
dialer load-threshold 255 balanced
!
interface Dialer0
ip address 99.0.0.2 255.0.0.0
encapsulation ppp
dialer in-band
dialer idle-timeout 500
dialer map ip 99.0.0.1 name atlanta broadcast 81012345678901
dialer load-threshold 255 balanced
dialer-group 1
ppp authentication chap
ppp multilink
```

MMP Examples

The examples in this section show MMP configuration without and with dialers.

Multichassis Multilink PPP without Dialers

The following example shows the configuration of MMP when no dialers are involved. Comments in the configuration discuss the commands. Variations are shown for a Cisco AS5200 access server or Cisco 4000 series router, and for an E1 controller.


```
! First make sure the multilink global virtual template number is defined on each
¡ stack group member.
multilink virtual-template 1

! If you have not configured any dialer interfaces for the physical interfaces in
! question (PRI, BRI, async, sync serial etc), you can define a virtual template.

interface virtual-template 1
ip unnumbered e0
ppp authentication chap
ppp multilink

! Never define a specific IP address on the virtual template because projected Virtual
! Access Interfaces are always cloned from the Virtual template interface. If a
! subsequent PPP link also gets projected to a stack member with a Virtual Access
! interface already cloned and active, we will have identical IP addresses on the two
! Virtual Interfaces. IP will erroneously route between them.

! On a AS5200 or 4X platform:

! On a TI controller
!
controller T1 0
framing esf
linecode b8zs
pri-group timeslots 1-24
!
interface Serial 0:23
no ip address
encapsulation ppp
no ip route-cache
ppp authentication chap
ppp multilink
!
! Or on an E1 Controller
!
controller E1 0
framing crc4
linecode hdb3
pri-group timeslots 1-31

interface Serial 0:15
no ip address
encapsulation ppp
no ip route-cache
ppp authentication chap
ppp multilink
```

Multichassis Multilink PPP with Dialers

When dialers are configured on the physical interfaces, do not specify the **ip unnumbered e0** on the virtual template interface. In this case, the virtual access interface acts as a passive interface, buttressed between the dialer interface and the physical interfaces associated with the dialer interface.

```
multilink virtual-template 1

interface virtual-template 1
ppp authentication chap
ppp multilink
```

```
! On a AS5200 or 4X platform:
!
interface dialer 1
ip unnum e0
dialer map .....
encap ppp
ppp authentication chap
dialer-group 1
dialer rotary 1

! On a T1 controller

controller T1 0
framing esf
linecode b8zs
pri-group timeslots 1-24
interface Serial0:23
no ip address
encapsulation ppp
dialer in-band
dialer rotary group 1
dialer-group 1
no ip route-cache
ppp authentication chap
ppp multilink
```

PPP Callback Example

The following example configures a PPP callback server and client to call each other.

The PPP callback server is configured on an ISDN BRI interface in a router in Atlanta. The callback server requires an enable timeout and a map class to be defined.

The PPP callback client is configured on an ISDN BRI interface in a router in Dallas. The callback client does not require an enable timeout and a map class to be defined.

PPP Callback Server

```
interface BRI0
 ip address 7.1.1.7 255.255.255.0
 encapsulation ppp
 dialer callback-secure
 dialer enable-timeout 2
 dialer map ip 7.1.1.8 name atlanta class dial1 81012345678901
 dialer-group 1
 ppp callback accept
 ppp authentication chap
!
map-class dialer dial1
 dialer callback-server username
```

PPP Callback Client

```
interface BRI0
 ip address 7.1.1.8 255.255.255.0
 encapsulation ppp
 dialer map ip 7.1.1.7 name dallas 81012345678902
 dialer-group 1
 ppp callback request
 ppp authentication chap
```


VPDN Examples

This section provides three examples that illustrate the following:

- One network access server (NAS) servicing multiple domains on multiple home gateways
- One NAS servicing multiple domains on one home gateway
- One NAS Using TACACS+ for Forwarding

Network Access Server Servicing Multiple Domains

This example provides VDPN configurations for a single network access server (NAS) and two different gateways. The two gateways are presumably located at two entirely separate companies. The NAS decides which company to forward to based on the domain name that is passed by the user.

The commands also illustrate where to configure the commands **vpdn outgoing** (on the network access server) and **vpdn incoming**(on a home gateway).

NAS1

```
vpdn enable
vpdn outgoing domain1.com nas1 ip 1.1.1.1
vpdn outgoing domain2.com nas2 ip 2.2.2.2
```

Gateway1—Domain1

```
vpdn enable
vpdn incoming nas1 gateway1 virtual-template 1

int virtual-template 1
  ip unnumbered Ethernet0
  ppp authentication chap
```

Gateway2—Domain2

```
vpdn enable
vpdn incoming nas2 gateway2 virtual-template 1

int virtual-template 1
  ip unnumbered Ethernet0
  ppp authentication chap
```

NAS Servicing Multiple Domains to the Same Gateway

This exmple provides configurations for one NAS and one Gateway that might have two parallel tunnels between them. Two different domain names are associated with two different virtual interface configurations.

Users dialing in with domain name “domain1.com” will be forwarded to the home gateway and be given a virtual-access interface based on virtual template 1. Users dialing in with the “domain2.com” will be fowarded to the same home gateway and be given a virtual-access interface based on virtual template 2.

NAS 1

```
vpdn enable
vpdn outgoing domain1.com nas1 ip 1.1.1.1
vpdn outgoing domain2.com nas2 ip 1.1.1.1
```

Gateway 1

```
vpdn incoming nas1 gateway virtual-template 1
vpdn incoming nas2 gateway virtual-template 2

interface virtual-template 1
  ip unnumbered Ethernet0
  peer default ip address pool domain1-pool
  ppp authentication chap

interface virtual-template 2
  ip unnumbered Ethernet0
  peer default ip address pool domain2-pool
  ppp authentication chap
```

Using TACACS+ for Forwarding from the NAS

This example provides configurations for an NAS and a public domain TACACS+ server. On the NAS it is only necessary to enable AAA and to use the **vpdn enable** command.

Users with structured logons (“user@domain.com”) will have their domain authorized on the TACACS server and will be forwarded if there is a VPDN entry there. If there is no VPDN entry on the TACACS server, the login process will continue as normal.

NAS

```
aaa new-model
vpdn enable
```

TACACS+ Server

```
vpdn outgoing domain.com nas ip 172.21.9.18
```